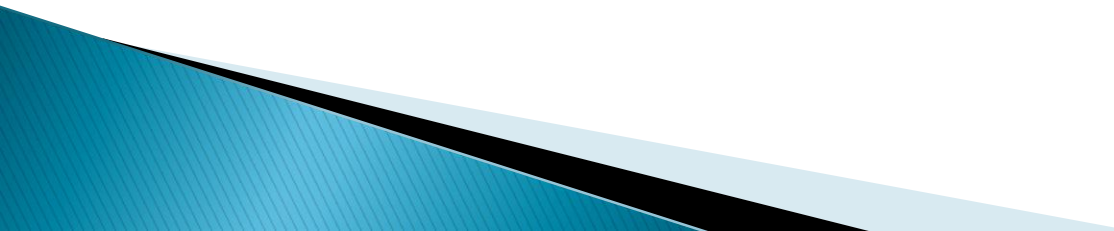


# SHA 512

Dr. Ditipriya Sinha  
C.S.E Department, NIT Patna

# Overview

- ▶ SHA-512 variant of SHA-256 which operates on eight 64-bit words.
  - ▶ The message to be hashed is first:
    - Padded with its length in such a way that the result is a multiple of 1024 bits long, and then
    - Parsed into 1024-bit message blocks  $M(1)$ ,  $M(2)$ , .....,  $M(N)$ .
- 

# Overview

- ▶ The message blocks are processed one at a time: Beginning with a fixed initial hash value  $H^{(0)}$ , sequentially compute

$$H^{(i)} = H^{(i-1)} + C_{M(i)}(H^{(i-1)})$$

- ▶ Here,  $C$  is the SHA-512 compression function and  $+$  means word-wise mod  $2^{64}$  addition.  $H(N)$  is the hash of  $M$ .

# Description of SHA-512

- ▶ The SHA-512 compression function operates on a 1024-bit message block and a 512 bit intermediate hash value.
- ▶ It is essentially a 512-bit block cipher algorithm which encrypts the intermediate hash value using the message block as key.
- ▶ Hence there are two main components to describe:
  - SHA-512 compression function, and
  - SHA-512 message schedule.

# Description of SHA-512

- ▶ Following notations will be used :

Notations	Meaning
$\oplus$	Bit wise XOR
$\wedge$	Bit wise AND
$\vee$	Bit wise OR
$+$	Mod $2^{32}$ addition
$R^n$	Right shift by n bits
$S^n$	Left shift by n bits

- ▶ All of these operators act on 64-bit words.

# Description of SHA-512

- ▶ The initial hash value  $H(0)$  is the following sequence of 64-bit words (which are obtained by taking the fractional parts of the square roots of the first eight primes).

$$H^{(0)}_1 = 6a09e667f3bcc908$$

$$H^{(0)}_2 = bb67ae85\ 84caa73b$$

$$H^{(0)}_3 = 3c6ef372fe94f82b$$

$$H^{(0)}_4 = a54ff53a5f1d36f1$$

$$H^{(0)}_5 = 510e527f\ ade682d1$$

$$H^{(0)}_6 = 9b05688c\ 2b3ecc1f$$

$$H^{(0)}_7 = 1f83d9abfb41bd6b$$

$$H^{(0)}_8 = 5be0cd19137e2179$$

# Description of SHA-512 (Preprocessing)

- ▶ Computation of the hash of a message begins by preparing the message:
  1. Pad the message in the usual way: Suppose the length of the message  $M$ , in bits, is  $l$ . Append the bit “1” to the end of the message, and then  $k$  zero bits, where  $k$  is the smallest non-negative solution to the equation  $l+1+k = 896 \bmod 1024$ .
  2. To this append the 128-bit block which is equal to the number  $l$  written in binary.

# Description of SHA-512 (Preprocessing)

► For example,

- The (8-bit ASCII) message “abc” has length  $8 \times 3 = 24$  so it is padded with a one, then  $896 - (24 + 1) = 871$  zero bits, and then its length to become the 1024-bit padded message 512-bit padded message.

01100001 01100010 01100011 1  $\underbrace{00 \dots 0}_{871}$   $\underbrace{00 \dots 0011000}_{128}$ .

- The length of the padded message should now be a multiple of 1024 bits.



# Description of SHA-512 (Preprocessing)

3. Parse the message into  $N$  1024-bit blocks  $M^{(1)}$ ;  $M^{(2)}$ ; ...;  $M^{(3)}$ . The first 64 bits of message block  $i$  are denoted  $M^{(i)}_1$ , the next 32 bits are  $M^{(i)}_2$ , and so on up to  $M^{(i)}_{15}$ .
- Note: The big-endian convention throughout, so within each 64-bit word, the left-most bit is stored in the most significant bit position.

# Description of SHA-256 (Algorithm)

For  $i = 1$  to  $N$  ( $N$  = number of blocks in the padded message)

{

- Initialize registers  $a, b, c, d, e, f, g, h$  with the  $(i-1)^{\text{st}}$  intermediate hash value (= the initial hash value when  $i = 1$ ) •

$$a \leftarrow H_1^{(i-1)}$$

$$b \leftarrow H_2^{(i-1)}$$

$\vdots$

$$h \leftarrow H_8^{(i-1)}$$

- Apply the SHA-512 compression function to update registers  $a, b, \dots, h$  •

For  $j = 0$  to 79

{

Compute  $Ch(e, f, g)$ ,  $Maj(a, b, c)$ ,  $\Sigma_0(a)$ ,  $\Sigma_1(e)$ , and  $W_j$  (see Definitions below)

$$T_1 \leftarrow h + \Sigma_1(e) + Ch(e, f, g) + K_j + W_j$$

$$T_2 \leftarrow \Sigma_0(a) + Maj(a, b, c)$$

$$h \leftarrow g$$

$$g \leftarrow f$$

$$f \leftarrow e$$

# Description of SHA-512 (Algorithm)

$$e \leftarrow d + T_1$$

$$d \leftarrow c$$

$$c \leftarrow b$$

$$b \leftarrow a$$

$$a \leftarrow T_1 + T_2$$

}

- Compute the  $i^{\text{th}}$  intermediate hash value  $H^{(i)}$  •

$$H_1^{(i)} \leftarrow a + H_1^{(i-1)}$$

$$H_2^{(i)} \leftarrow b + H_2^{(i-1)}$$

$\vdots$

$$H_8^{(i)} \leftarrow h + H_8^{(i-1)}$$

}

$H^{(N)} = (H_1^{(N)}, H_2^{(N)}, \dots, H_8^{(N)})$  is the hash of  $M$ .

# Definition

- ▶ Six logical functions are used in SHA-512.
- ▶ Each of these functions operates on 64-bit words and produces a 64-bit word as output. Each function is defined as follows:

$$Ch(x, y, z) = (x \wedge y) \oplus (\neg x \wedge z)$$

$$Maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$

$$\Sigma_0(x) = S^{28}(x) \oplus S^{34}(x) \oplus S^{39}(x)$$

$$\Sigma_1(x) = S^{14}(x) \oplus S^{18}(x) \oplus S^{41}(x)$$

$$\sigma_0(x) = S^1(x) \oplus S^8(x) \oplus R^7(x)$$

$$\sigma_1(x) = S^{19}(x) \oplus S^{61}(x) \oplus R^6(x)$$

# Definition

- ▶ Expanded message blocks  $W_0; W_1; \dots; W_0$  are computed as follows via the SHA-512 message schedule:

$W_j = M^{(i)}_j$  for  $j = 0, 1, \dots, 15$ , and

For  $j = 16$  to  $63$

{

$$W_j \leftarrow \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16}$$

}

# Definition

- ▶ A sequence of constant words, K0; ...; K79; is used in SHA-256. In hex, these are given by

```
428a2f98d728ae22 7137449123ef65cd b5c0fbcfec4d3b2f e9b5dba58189dbbc
3956c25bf348b538 59f111f1b605d019 923f82a4af194f9b ab1c5ed5da6d8118
d807aa98a3030242 12835b0145706fbe 243185be4ee4b28c 550c7dc3d5ffb4e2
72be5d74f27b896f 80deb1fe3b1696b1 9bdc06a725c71235 c19bf174cf692694
e49b69c19ef14ad2 efbe4786384f25e3 0fc19dc68b8cd5b5 240ca1cc77ac9c65
2de92c6f592b0275 4a7484aa6ea6e483 5cb0a9dc41fbd4 76f988da831153b5
983e5152ee66dfab a831c66d2db43210 b00327c898fb213f bf597fc7beef0ee4
c6e00bf33da88fc2 d5a79147930aa725 06ca6351e003826f 142929670a0e6e70
27b70a8546d22ffc 2e1b21385c26c926 4d2c6dfc5ac42aed 53380d139d95b3df
650a73548baf63de 766a0abb3c77b2a8 81c2c92e47edaee6 92722c851482353b
a2bfe8a14cf10364 a81a664bbc423001 c24b8b70d0f89791 c76c51a30654be30
d192e819d6ef5218 d69906245565a910 f40e35855771202a 106aa07032bbd1b8
19a4c116b8d2d0c8 1e376c085141ab53 2748774cdf8eeb99 34b0bcb5e19b48a8
391c0cb3c5c95a63 4ed8aa4ae3418acb 5b9cca4f7763e373 682e6ff3d6b2b8a3
748f82ee5defb2fc 78a5636f43172f60 84c87814a1f0ab72 8cc702081a6439ec
90befffa23631e28 a4506cebd82bde9 bef9a3f7b2c67915 c67178f2e372532b
ca273ecdea26619c d186b8c721c0c207 eada7dd6cde0eb1e f57d4f7fee6ed178
06f067aa72176fba 0a637dc5a2c898a6 113f9804bef90dae 1b710b35131c471b
28db77f523047d84 32caab7b40c72493 3c9ebe0a15c9bebc 431d67c49c100d4c
4cc5d4becb3e42b6 597f299cfc657e2a 5fcb6fab3ad6faec 6c44198c4a475817.
```

- ▶ These are the first 64 bits of the fractional parts of the cube roots of the first 80 primes.

# Diagram

The SHA-512 compression function is pictured below:

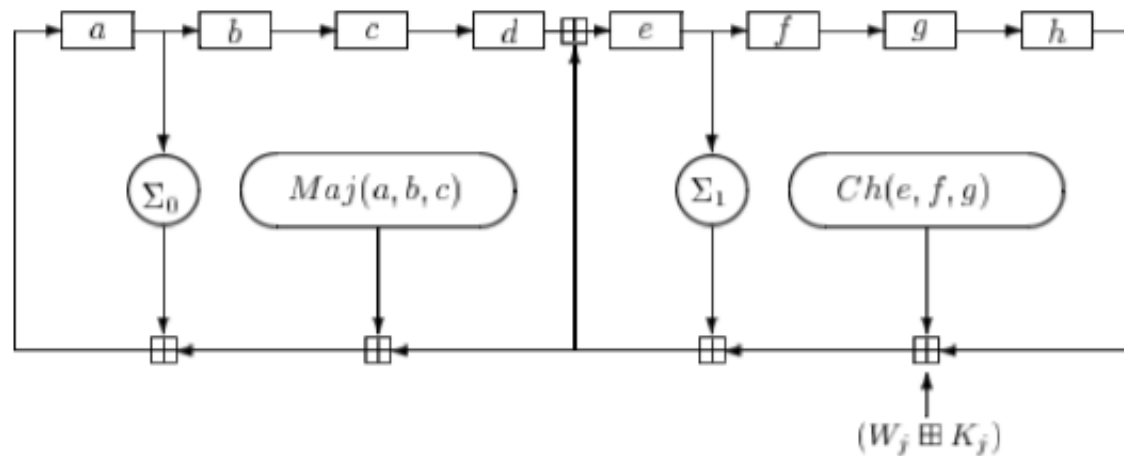
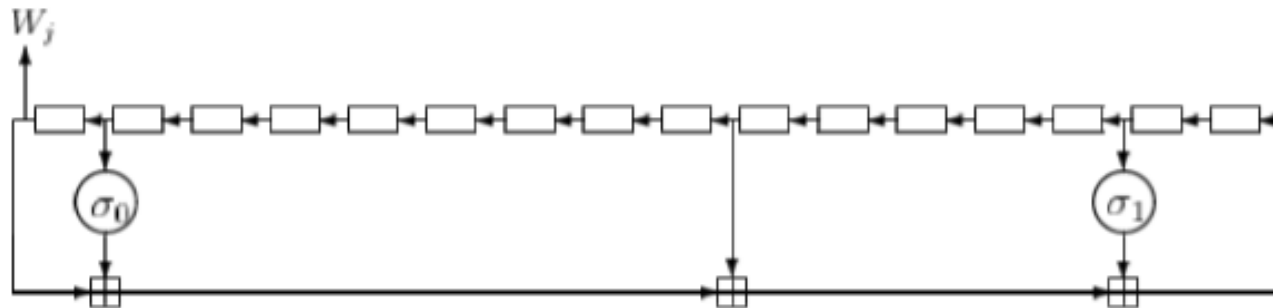


Figure 3:  $j^{\text{th}}$  internal step of the SHA-512 compression function  $C$

where the symbol  $\boxplus$  denotes mod  $2^{64}$  addition.

# Diagram

The message schedule can be drawn as follows:



**Figure 4:** SHA-512 message schedule

The registers here are loaded with  $W_0, W_1, \dots, W_{15}$ .



# Hash value of “abc”

- ▶ The result of hashing the 24-bit message "abc". After padding the message becomes (in hexadecimal) :

```
61626380 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000018
```

- And the hash value is :

ddaf35a193617abacc417349ae20413112e6fa4e89a97ea20a9eeee64b55d39a  
2192992a274fc1a836ba3c23a3feebbd454d4423643ce80e2a9ac94fa54ca49f

- ▶ The result of hashing the 448-bit message :

“abcdefghijklmnopqrstuvwxyz  
Hijklmnopqrstuvwxyz”

# Hash value of “abc”

- ▶ Which, after padding, becomes the 2-block message:

```
61626364 65666768 62636465 66676869 63646566 6768696a 64656667 68696a6b
65666768 696a6b6c 66676869 6a6b6c6d 6768696a 6b6c6d6e 68696a6b 6c6d6e6f
696a6b6c 6d6e6f70 6a6b6c6d 6e6f7071 6b6c6d6e 6f707172 6c6d6e6f 70717273
6d6e6f70 71727374 6e6f7071 72737475 80000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000380.
```

- ▶ The hash value for this message is :

```
8e959b75dae313da8cf4f72814fc143f8f7779c6eb9f7fa17299aeadb6889018
501d289e4900f7e4331b99dec4b5433ac7d329eeb6dd26545e96e55b874be909
```